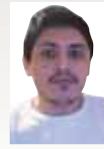


# CONTROL DE VERSIONES USANDO SUBVERSION



DISEÑADOR WEB, APASIONADO-  
POR TODO LO QUE VAYA DESPUÉS  
DE HTTP://

EN UN AMBIENTE DE TRABAJO UNIPERSONAL, ES SENCILLO CONTROLAR EL VERSIONADO DE UN DOCUMENTO XHTML. PERO ¿QUÉ PASA CUANDO TRABAJAMOS EN EQUIPO? LES PRESENTAMOS SUBVERSION, UNA HERRAMIENTA PARA FACILITAR LA COORDINACIÓN ENTRE GRUPOS DE TRABAJO.

▼ Seguramente, en algún momento de nuestra carrera profesional nos hemos preguntado lo siguiente: “¿Quién tiene la última versión de este archivo XHTML?”, “¿cuáles son los últimos cambios que se hicieron en las hojas de estilo?”, “¿cómo puedo evitar sobreescribir el código fuente de mi compañero?”. Si no tenemos un sistema confiable para mantener y coordinar versiones de código, responder a estas preguntas es casi imposible, y trabajar a ciegas puede ser el único camino posible.

En esta edición hablaremos de Subversion, tal vez el sistema más utilizado para controlar versiones de documentos, ya sean éstas

código fuente, páginas o documentación. El artículo estará enfocado en un caso inicial, pero práctico, de utilización del software, y dejaremos los detalles puntiagudos a la curiosidad del lector.

## PRESENTANDO SUBVERSION

Como dijimos anteriormente, Subversion (SVN) es un sistema de control de versiones: **maneja los archivos y las carpetas de un proyecto y sus modificaciones en el transcurso del tiempo.** Fue creado en el año 2000 en CollabNet Inc, con el objetivo de ser una alternativa real y mejorada a CVS (*Concurrent Versions System*, un sistema de control de versiones que era la solución de facto en la época). Al igual que otros proyectos exitosos de renombre, SVN es libre y Open Source, ya que está distribuido bajo la licencia Apache.

**Subversion trabaja replicando el modelo cliente/servidor:** uno o más clientes se conectan a un servidor central que tiene la última copia del proyecto, como también copias de sus versiones anteriores, dentro de lo que llamaremos un **repositorio**. El cliente pide la última versión disponible del proyecto para poder trabajar. Los clientes hacen cambios y, por último, sincronizan sus copias con el repositorio.

Como pueden observar, su uso en equipos de trabajo está perfectamente justificado. De cualquier manera, Subversion también es útil como una herramienta de backup incremental unipersonal. Imaginemos SVN como un sistema de históricos con capturas en el código fuente automáticas. Cada vez que enviemos un archivo modificado al repositorio, éste creará una copia de la versión anterior. Ninguna de las copias anteriores se perderá, por lo que siempre podremos volver a una versión anterior de nuestros archivos si es necesario. Es más, si estamos trabajando con un servidor de Subversion ubicado en algún lugar de Internet, digamos, un servidor alojado en los Estados Unidos, tenemos una buena posibilidad de no perder los archivos originales por un desperfecto en nuestras computadoras. Como introducción, en este informe solamente nos conectaremos a un repositorio creado en nuestra propia máquina, pero en el recuadro Subversion en la Web pueden encontrar servidores alojados de Subversion listos para probar.

## LO BÁSICO: CHECKOUT, UPDATE Y COMMIT

Los tres comandos que usaremos con más frecuencia al trabajar con Subversion son **svn checkout**, **svn update** y **svn commit**.

## Subversion en la Web

Tenemos la suerte de que, últimamente, vienen apareciendo servicios interesantes basados en la Web para probar Subversion. Algunos ofrecen cuentas gratuitas, limitadas en su funcionalidad, claro, y otros están destinados a un uso fuerte del sistema. Los siguientes son los más interesantes:

### Beanstalk ([www.beanstalkapp.com](http://www.beanstalkapp.com))

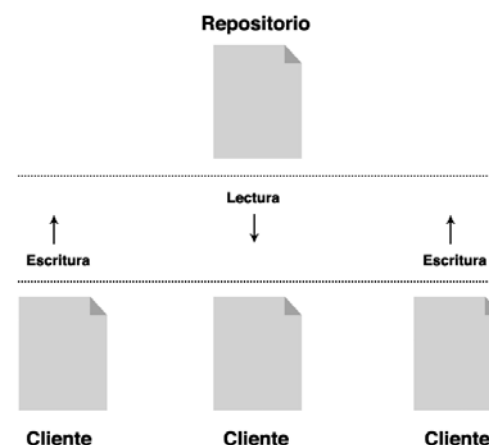
Servicio nuevo y con buena interfaz, realmente muy fácil de usar. Es gratuito con hasta 20 MB de almacenamiento. Se destaca su excelente soporte. Es ideal para los que recién comienzan con Subversion.

### Roundhaus ([www.roundhaus.com](http://www.roundhaus.com))

Aunque la cuenta gratuita tiene sus limitaciones, se destaca por ser uno de los pocos que ofrecen un servicio de integración continua, lo cual lo convierte en algo más que un simple repositorio en la Web.

### Google Code (<http://code.google.com>)

Dentro del amplio abanico de servicios gratuitos de Google, se destaca Google Code, que ofrece hosting gratuito ilimitado con acceso, claro, a Subversion.



**Figura 1.**  
La representación gráfica en la que podemos ver un repositorio con tres clientes conectados. Subversion ordena y facilita el trabajo.

## FOROS

[www.redusers.com/foros/webmasters](http://www.redusers.com/foros/webmasters)

Veamos en qué consisten:

### 1) SVN CHECKOUT

Descarga una copia privada de la última versión del proyecto, es decir, una *working copy*, a nuestra máquina. Es lo primero que tenemos que hacer al iniciar una sesión de trabajo con SVN.

### 2) SVN UPDATE

Como su nombre indica, se usa para actualizar nuestra *working copy* con los últimos cambios hechos por nuestros compañeros. Este comando debe ejecutarse a menudo para tener siempre una copia fresca del proyecto.

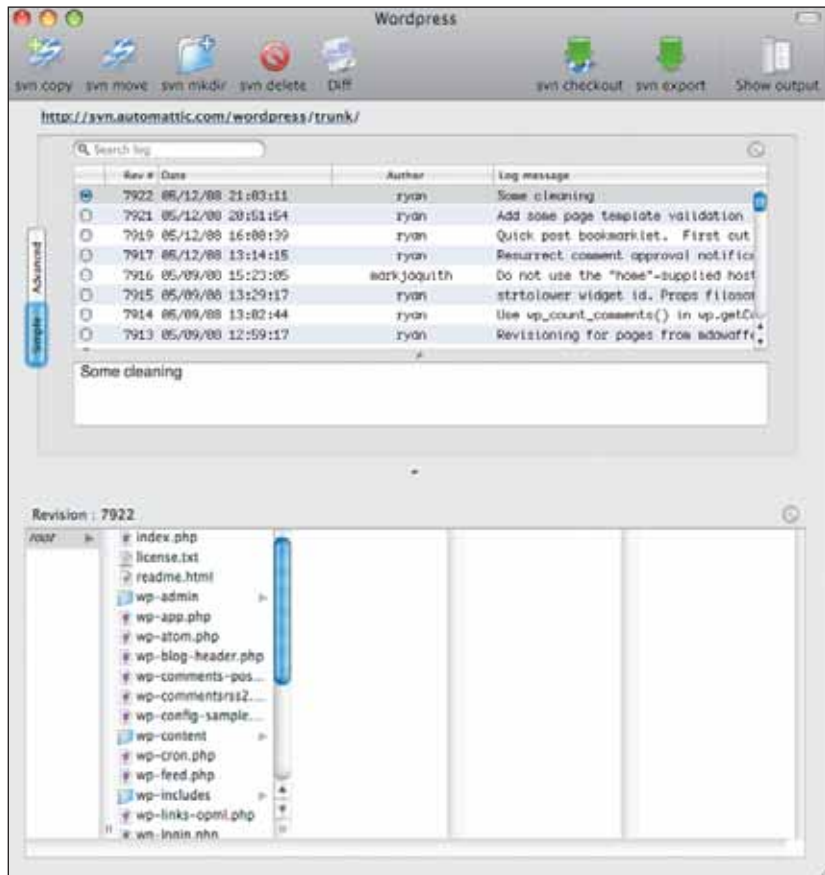
### 3) SVN COMMIT

Envía al repositorio los cambios que hicimos en nuestra copia privada. De más está decir que es una buena práctica, en cada **commit**, comentar los cambios que hicimos en la versión que estamos enviando. De esta manera, nuestros compañeros de equipo tendrán una buena ayuda sobre qué es lo nuevo en la última versión que están descargando (los comentarios se pueden agregar usando el mismo comando **commit** o, como veremos más adelante, completando un campo en una ventana).

No se preocupen: todos estos comandos tienen sus semejantes en la herramienta visual que usaremos en este artículo, TortoiseSVN, por lo que haciendo un par de clics evitaremos tipearlos en una ventana de terminal. Sin embargo, lo importante en este punto es que entiendan el proceso de trabajar con Subversion, que en su concepción más simple consiste en tres pasos: 1. **checkout**, 2. **update** y 3. **commit**. Con el tiempo aprenderemos más comandos útiles para trabajar con el sistema, como **svn diff**, que permite buscar diferencias entre los archivos que estamos subiendo.

## CONECTANDO AL REPOSITORIO DE WORDPRESS

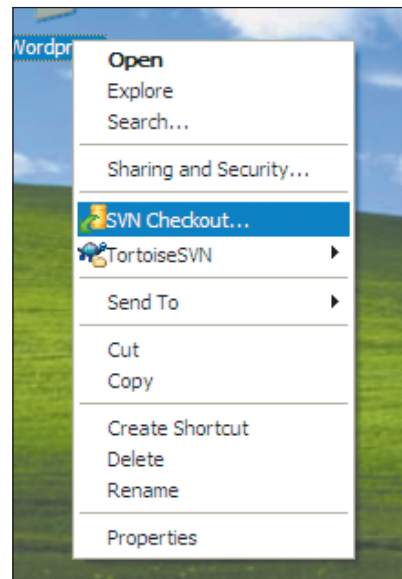
Vamos a ver un ejemplo rápido para familiarizarnos con la herramienta: nos conectaremos al repositorio (de sólo lectura) de WordPress. Como muchos saben, WordPress es el CMS para *weblogs* más utilizado en la actualidad. Aunque se puede descargar directamente desde [www.wordpress.org](http://www.wordpress.org), lo que a



**Figura 2.** svnX, el cliente más popular en Mac, conectado a un repositorio. Fíjense en todas las versiones del proyecto (columna Rev #) y sus comentarios (columna Log Message).

nosotros nos interesará por el momento son sus versiones nocturnas, públicamente disponibles mediante SVN.

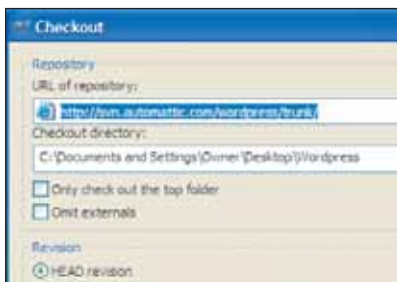
Primero, debemos bajar TortoiseSVN (gratis, Open Source) de <http://tortoisesvn.net/downloads>. TortoiseSVN viene en un archivo instalable .msi, y en versiones de 32 y 64 bits. En este paso también podemos bajar el paquete del idioma español. Nota: los usuarios de Mac tenemos en svnX el cliente perfecto para usar Subversión, que se puede descargar de <http://rubyurl.com/RGbz>. Todos los comandos de TortoiseSVN tienen sus equivalentes en svnX, por lo que a los lectores "maqueros" les será sencillo adaptar las instrucciones de este artículo. Ahora, una vez descargada e instalada la herramienta en la PC, nos conectamos al servidor SVN de WordPress. Para esto, crearemos una nueva carpeta en alguna parte de nuestra PC que funcionará como *working copy*, o copia privada del proyecto en que se trabaja.



**Figura 3.** Un menú contextual de Windows con ítems agregados por TortoiseSVN.

## WEBMASTERS

### COORDINANDO VERSIONES DE CÓDIGO FÁCILMENTE



**Figura 4. La ventana para hacer checkout.**

Hacemos clic derecho en esa carpeta y Windows, además de las opciones habituales de un menú contextual, nos mostrará dos opciones propias de la herramienta: **[SVN Checkout]** y **[TortoiseSVN]** (Figura 3).

Como ven, TortoiseSVN no tiene una ventana de aplicación propiamente dicha, funciona solamente como un agregado al menú contextual de Windows, lo cual le otorga una gran flexibilidad.

Del menú contextual elegimos la opción **[SVN Checkout]**, y TortoiseSVN nos mostrará una ventana para ingresar datos del servidor SVN (Figura 4). Escribimos <http://svn.automattic.com/wordpress/trunk> en el campo **[URL of repository]** y hacemos clic en **[OK]** para empezar a ejecutar el comando. TortoiseSVN copiará la última versión del repositorio de WordPress a esa carpeta. En este paso también es interesante la opción **[Show log]**, que despliega un historial con las versiones anteriores a la actual y sus comentarios.

Eso es todo, ya podemos disfrutar de la última versión de WordPress en una carpeta de nuestra PC. Veamos ahora cómo crear un repositorio y escribir en él.

### CREANDO UN REPOSITORIO

Una de las ventajas de TortoiseSVN es que facilita la creación de repositorios locales. Literalmente, con un clic derecho en una carpeta determinada se puede crear un repositorio en segundos. Veamos un ejemplo completo. Los pasos que debemos hacer son los siguientes:

#### 1) CREAR EL REPOSITORIO

Creamos una nueva carpeta en alguna parte de nuestra PC. Hacemos clic derecho sobre ella y buscamos la opción **[Create repository here]**, que se encuentra dentro del ítem **[TortoiseSVN]**. La herramienta nos mostrará una ventana donde tendremos que elegir el tipo de repositorio. Elegimos **[Native filesystem]** (sistema de archivos nativo) y continuamos. TortoiseSVN creará la estructura de carpetas necesaria para el funcionamiento de Subversion.

#### 2) HACER EL PRIMER CHECKOUT

Ahora debemos crear una carpeta para trabajar en nuestra copia privada y hacer el primer *checkout*.

Para esto, seguimos los mismos pasos que en el ejemplo anterior (conectando a WordPress), pero esta vez, en vez de apuntar al repositorio de WordPress, apuntamos a la carpeta de nuestro repositorio creada en el punto anterior. TortoiseSVN descargará la versión 0 del repositorio, es decir, no descargará ningún archivo.

#### 3) HACER ALGÚN CAMBIO

Ahora bien, creemos un archivo nuevo de texto en la misma carpeta del punto anterior. Este archivo será luego nuestro primer *commit*.

#### 4) HACER EL PRIMER COMMIT.

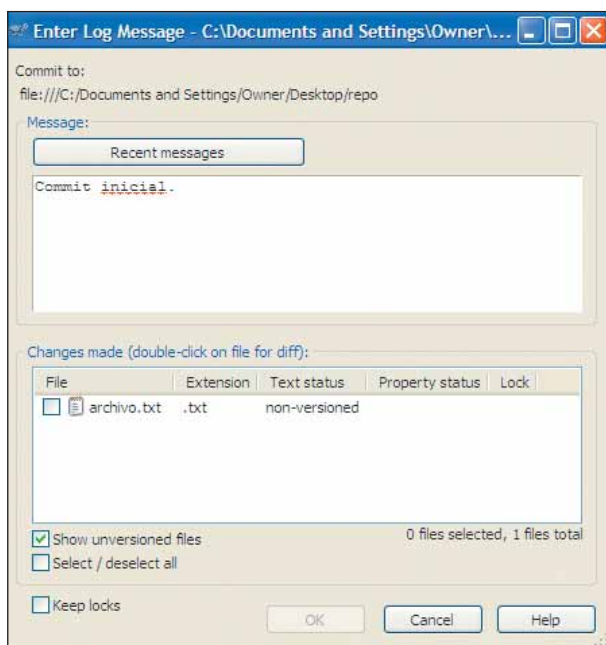
Ahora subiremos el archivo creado en el punto anterior. Si prestamos atención, el icono de la carpeta con nuestra copia privada ahora tiene un icono de una tilde. Esto quiere decir que hay cambios en esa carpeta, listos para ser sincronizados en el repositorio. Hacemos clic derecho en la carpeta y seleccionamos **[SVN Commit]**. TortoiseSVN mostrará una ventana con información que tendremos que completar sobre la versión que estamos subiendo. En la mitad inferior de esta ventana veremos los archivos que fueron modificados (en este caso, el archivo de texto) y sus propiedades. Seleccionamos el archivo y continuamos. TortoiseSVN aplicará los cambios en el repositorio para completar la flamante versión 1 del proyecto (Figura 5).

De aquí en más, el proceso de trabajo será similar: hacer un checkout de la última versión en el repositorio, realizar cambios en nuestra copia privada, luego hacer un *update* y, por último, un *commit*. En la práctica, y con grupos de trabajo, el proceso no es tan sencillo, ya que si hay inconsistencias entre los archivos, tendremos que combinarlos (lo que en la jerga se conoce como *merge*), lo cual a veces puede resultar en más de un dolor de cabeza. Pero, claro, todo esto se aprende durante la marcha.

### CONCLUSIÓN

Subversion puede parecer difícil, pero una vez que se le pierde el miedo, es una herramienta sumamente útil. Aunque tal vez SVN sea más conocido en el ámbito del desarrollo de software a gran escala, ya es hora de que, como desarrolladores web, tengamos a mano una herramienta de control de versiones, aun cuando trabajemos solos.

Esperamos que se hayan entusiasmado con Subversion, y como siempre, el foro de USERS está abierto para cualquier consulta. ▲



**Figura 5. Aquí estamos haciendo commit de un archivo en el software gratuito y open source TortoiseSVN.**